**Analysis & Understanding for Theatre Seating Problem**:

At first I went through the problem statement and the inputs provided in the document to clearly understand what the problem is about. Then I further proceeded with writing the below steps / pseudo code in a notepad to attain the solution based on the rules/conditions provided in the document.

1. Identify the total seats count available in the Theatre
2. Parse each customer seat requests from the input provided

For each customer/seat request , will check

**if Requested seats < Total Seats available in Theatre then**

1. Store the Rows and Sections/Columns with Seat count in a list “**seatLayoutsSelection**” by checking if the Sectionseatcount in theatre is greater or equal to the requested seats. Will be using this list to determine / assign the seats to the parties
2. if the list **seatLayoutsSelection > 0** then

* then filter the start row(i.e, Row 0) from the seatLayoutSelections list
* Now will need to consider the next row after the start row and for that will look for the next minimum row after the start row

(the above filters are considered as per requirement that says “*Put parties as close to the front as possible.*”)

* Now look at both of these above 2 rows rows to see if there are any exact matches between the section seats count and the customer seat requests count.
* If there are any exact match then
  + Assign the seats to customer with the respective section in that row
  + Reduce the section seats count and the count of the assigned section in that row should be zero now.

if there is not an exact match between section seat count and the customer seat count (ie., the customer seats count is lesser than the section count)

* + Assign the seats to customer from the sections from first available row
  + Reduce the section seats count and the count of the assigned section should be zero now.

1. **If seatLayoutsSelection < 0 then**

it means that requested seats count for a single person is not available at any single section. So we will be telling them “*Call to split party”*

**if** **Requested seats > Total Seats available in Theatre then**

1. We will tell to customer “Sorry, we can’t handle your party”

**Challenges in achieving the above solution:**

Initially the problem statement was not clear and so later I have analysed the problem statement based on the inputs provided against it.

While implementing the solution, I was bit confused initially to apply the logic for picking the first two rows based on the seating priority. After some time of research, I found some result to achieve the logic through the LINQ query. Then I was able to get the expected results.

**Strengths:**

1. With respect to this problem , I would say my strength is instead of thinking about the final output, I have first tried to achieve the intermediate solutions. For example: In this problem , I slowly started from identifying total seats in the theatre and compared with the customer request. Then I continued with achieving the next expected intermediate result and followed the same process until I achieved the final expected output.
2. I have learnt some new features in LINQ with the use of Skipwhile method and I will use this somewhere in the project when needed

**Weakness:**

1. It would have been difficult or would have taken long time to solve this problem with out researching on the internet about LINQ features.